Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Discrete Applied Mathematics 144 (2004) 263 – 269
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Room allocation: a polynomial subcase of the quadratic assignment problem

Valentina Ciriani, Nadia Pisanti, Anna Bernasconi

Dipartimento di Informatica, Università di Pisa, Via Buonarroti, 2, Pisa 56127, Italy

Received 8 May 2002; received in revised form 9 January 2004; accepted 22 January 2004

Abstract

The quadratic assignment problem (QAP) is among the hardest combinatorial optimization problems. Very few instances of this problem can be solved in polynomial time. In this paper we address the problem of allocating rooms among people in a suitable shape of corridor with some constraints of undesired neighborhood. We give a linear time algorithm for this problem that we formulate as a QAP.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Quadratic assignment problem; Allocation; Combinatorial algorithm

0. Introduction

The quadratic assignment problem (QAP) is among the hardest combinatorial optimization problems [4,7]. Indeed it is an NP-hard problem. Nevertheless, there are (very few) polynomial instances of it [1,2,8]. In this paper we address an instance of QAP, with the following interpretation. Assume an institution collects people subdivided into groups and organized into hierarchies, with the unavoidable dose of bad relations among groups. Suppose you get a new building and you must allocate people into rooms, avoiding that people that are in bad relation happen to be in the neighborhood of each other. This paper addresses this problem by formalizing it, and providing a linear time algorithm that optimally solves it. More precisely our linear time solution solves the problem when the number of rooms is equal to the number of persons, or there is at most one empty room. Surprisingly, this linearly solvable optimization problem is an instance of the QAP. Notice that any allocation problem with weighted costs associated to any relation (not necessarily neighborhood) can be viewed as an instance of the QAP. A preliminary version of this paper [3] also addressed further cases of increasing difficulty of this problem, and provided polynomial algorithms for them.

1. Preliminary definitions

Assume there are n persons $p_1, \ldots, p_n$. These persons have different positions within the institution, and their opinions count proportionally to their power. Hence, to each person p_i we associate a weight w_i, which is a positive integer. Rooms are arranged along a corridor of a suitable length, such that at each point of the corridor there is a room per side, and the two sides are symmetrical. Formally, a *corridor* of length m is a $2 \times m$ matrix $\mathbf{C}$. Each entry of the matrix represents a *room*. Without loss of generaility, we number the rooms as shown in Fig. 1, where we (arbitrarily) also show that the corridor is open on the left-hand side. We say that room $\mathbf{C}_{i,j}$ and $\mathbf{C}_{i',j'}$ are *neighbors* if $i = i'$ and $|j - j'| = 1$, or if $i \neq i'$ and $j = j'$. For example, in Fig. 1, the neighbors of room $\mathbf{C}_{1,j}$ are $\mathbf{C}_{2,j}$, $\mathbf{C}_{1,j-1}$, and $\mathbf{C}_{1,j+1}$. For a room $C_{i,j}$, we say that

E-mail addresses: ciriani@di.unipi.it (V. Ciriani), pisanti@di.unipi.it (N. Pisanti), annab@di.unipi.it (A. Bernasconi).

0166-218X/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2004.01.004

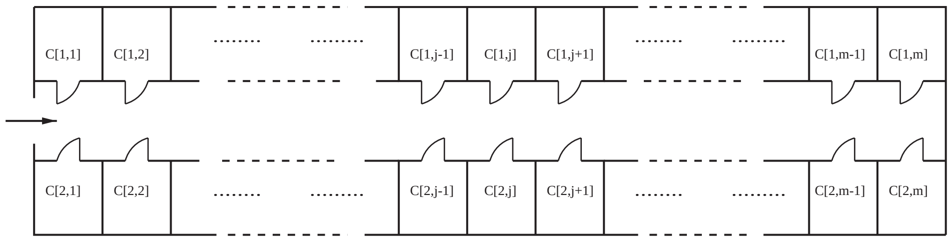

Fig. 1. A corridor.

it is in line i and column j. We assume that persons are partitioned into *families*. Given two persons p_i and p_j, we say that they are *compatible* if they belong to the same family. They are *incompatible* otherwise. The compatibility relation is reflexive, symmetric, and transitive. A *room allocation* ρ is assignment of rooms to people such that one person is assigned exactly one room.

Definition 1. If ρ assigns p_i and p_j to two neighbor rooms and they are not compatible, we say that there is a *crash* c in ρ. The weight of this crash is $w_c = w_i + w_j$.

Finally, we define the parameter that will measure the cost of a room allocation.

Definition 2. Given n people with their weights, and a room allocation ρ, we define the *cost* μ of the room allocation as the sum of the weights of all the crashes c in ρ, that is

$$\mu(\rho) = \sum_{c \in \rho} w_c.$$

The goal is to find the room allocation ρ that minimizes $\mu(\rho)$.

2. The QAP formalization

Let us assume that there are as many rooms as people, i.e., $n = 2m$.[1] The problem of room allocation, when there are no extra rooms, can be formulated in the following way, showing that it is a special case of a QAP (see [7,6,5]):

$$\min \sum_{i,j,h,k} X_{ij} X_{hk} Q_{ijhk},$$

$$\sum_{i=1}^{n} X_{ij} = 1 \quad \forall j = 1, \ldots, 2m,$$

$$\sum_{j=1}^{2m} X_{ij} = 1 \quad \forall i = 1, \ldots, n,$$

$$X_{ij} \in \{0, 1\},$$

where $X_{ij} = 1$ if room j is allocated to the person i, and $X_{ij} = 0$ otherwise. The constraints $\sum_i X_{ij} = 1$ and $\sum_j X_{ij} = 1$ ensure that there is a one-to-one correspondence between rooms and people. The matrix Q is defined in the following way, where N is the neighborhood relation (that is, $(k, j) \in N$ iff rooms j and k are neighbor):

$$Q_{ijhk} = \begin{cases} 0 & \text{if } (k,j) \notin N, \\ 0 & \text{if } (k,j) \in N \text{ and } i \text{ and } h \text{ belong to the same family}, \\ w_i + w_h & \text{if } (k,j) \in N \text{ and } i \text{ and } h \text{ do not belong to the same family}. \end{cases}$$

[1] The solution of the case $n = 2m - 1$, that is when n is odd and thus there is one empty room left, can be formulated as a QAP as well. Such formalization and a linear algorithm for the resulting allocation problem is described in Section 4.

3. The algorithm

3.1. A circular corridor

The corridor of Fig. 1 has two lateral borders, corresponding to the first and last column of the matrix C representing the rectangular corridor. The borders allow to avoid some conflicts and thus introduce some special cases that we will discuss in the next section. In this section, we give the intuition of our general strategy by considering first a *circular corridor* without lateral borders. Later, we will show how to preprocess such strategy in order to assign families to the borders, and solve the case of the rectangular corridor. We can observe that, since $\mu(\rho)$ is defined as the sum of single crashes, and since each crash is defined as the sum of the weights of the persons involved, we have that $\mu(\rho)$ is just the sum of the weights of all members of each family exposed to crashes, where the weight of a person is counted as many times as the number of crashes it is exposed to. Therefore, minimizing $\mu(\rho)$ equals minimizing the number and weights of crashes of each single family independently. Formally:

Proposition 1. *Let* $\mathbf{F}$ *be the set of all families, and let* μ_f *be sum of the weights of the members of* $f \in \mathbf{F}$ *that are exposed to crashes in a room allocation* ρ. *We have that*

$$\min \mu(\rho) = \sum_{f \in \mathbf{F}} \min \mu_f(\rho).$$

As a consequence, any solution that minimizes the cost of the crashes of each single family is clearly optimal (notice that, nevertheless, not all optimal solutions are composed of optimal local solutions). Hence, if we find a room allocation that minimizes such costs for each family, then we definitely have an optimal solution for the problem because we have matched the lower bound for $\mu(\rho)$. Let us denote with $\bar{\mu}$ the cost of such optimal solution. Thus, let us consider the problem of minimizing μ_f for each family $f \in \mathbf{F}$. First of all, we can observe that people belonging to the same family should be always placed close to each other in an optimal solution. This holds because they give no crash between each other, and hence their neighboring has zero cost. The exact way in which the family should be arranged depends on the parity of the number of members of f. Namely, we have that if a family has an even number of members $2n_1$, then its members should be arranged into a set of rooms in the shape of a rectangle:

$$\begin{array}{llll} C_{1,i} & C_{1,i+1} & \dots & C_{1,i+n_1-1} \\ \\ C_{2,i} & C_{2,i+1} & \dots & C_{2,i+n_1-1} \end{array}$$

where obviously in $\bmod m$ we will have that $0 \leqslant i$ and $i + n_1 \leqslant m$. In this case, there are at most four members of the family involved in exactly four crashes (in the case of size 2 families the member involved in the crashed are obviously just 2). Namely, the person in room $C_{1,i}$ crashes with the person in room $C_{1,i-1}$, the person in room $C_{2,i}$ crashes with the one in room $C_{2,i-1}$, and, similarly, the two right extremes of the family crash with the rooms with column index $i + n_1$, where all the column indexes above have to be considered $\bmod m$ operation (and of course $C_{x,0} = C_{x,m}$). It is straightforward to verify that any other arrangement of an even size family results in more than four crashes. For analogous reasons, if the family has an odd number $2n_1 + 1$ of members, then it should be arranged in the following shape:

$$\begin{array}{lllll} C_{1,i} & C_{1,i+1} & \dots & C_{1,i+n_1-1} & \\ \\ C_{2,i} & C_{2,i+1} & \dots & C_{2,i+n_1-1} & C_{2,i+n_1} \end{array}$$

(where we will refer to room $C_{2,i+n_1}$ as the *antenna*), or equivalently the symmetrical ones (with the antenna-room in line 1, and the specular cases on the left). In this case there are again at most four persons involved in crashes (except that there are three in the case of a size 3 family), but there are five crashes. In fact the person in the antenna-room $C_{2,i+n_1}$ crashes twice: one with the person in room $C_{2,i+n_1+1}$ and one with the person in room $C_{1,i+n_1}$ (again, the column indexes have to be considered $\bmod m$).

From now on we will consider the partition of $\mathbf{F}$ into the two sets $\mathbf{F}_{\text{odd}}$ and $\mathbf{F}_{\text{even}}$ containing families of odd and even size, respectively. An exhaustive check of all possible alternative shapes for the families shows that those above are the minimum cost ones for even and odd size families, respectively. Hence, the shapes above are those that minimize the number of possible crashes, and thus they represent the local optimal placements of the members of each family in $\mathbf{F}$. Applying Proposition 1, we can build our optimal solution with the families arranged as above. This observation is the key point of the linear algorithm we are going to suggest for this problem. Since n is an even number, there will be an even number of odd size families. Two of them together (with the antennae in front of each other in different lines of the corridor) result in an overall rectangular shape. What our algorithm actually does is to pair two by two all the odd size families. Observe that any other arrangement of two odd-sized families different from the rectangular one, leads to worse solutions. Thus, we propose the following general strategy:

Step 1: For each family f with at least four members, find the four members of minimal weights. Let them be f_1, f_2, f_3, f_4, and let $w_1^f \leqslant w_2^f \leqslant w_3^f \leqslant w_4^f$ be their weights. Hence

- let each $f \in \mathbf{F}_{\text{even}}$ be placed in the shape of a rectangle with f_1, f_2, f_3 and f_4 at its borders.
- let $f \in \mathbf{F}_{\text{odd}}$ be placed in the shape of a mobile telephone with f_1, f_2, f_3 and f_4 at its borders, and f_1 as antenna element.

The members of a family of size 2 are placed in two rooms $C_{1,j}$ and $C_{2,j}$. Finally, a family with 3 members has an "L" shape with the two least-weight members f_1 and f_2 at the two antenna positions, and the third element on the corner.

Moreover, odd size families are paired (remind that there is an even number of them) in such a way to place the antennae members in front of each other, and together form a rectangle shape. Therefore, we are actually left with rectangle shapes only.

Step 2: We place all the rectangle shapes resulting from *Step* 1 in any way.

Let ρ denote the allocation resulting from the two steps above. We now compute $\mu(\rho)$ as the sum of the $\mu_f(\rho)$ for each family f, which is minimal and that can be computed in the following way:

- For families $f \in \mathbf{F}$ with at least four members, denote with $f_{\text{tot}} = w_1^f + w_2^f + w_3^f + w_4^f$ the total weight of members exposed to crashes; moreover, if $f \in \mathbf{F}_{\text{odd}}$, denote with $f_{\min} = w_1^f$ the weight of the member that will be placed in the antenna-room, and thus that will crash twice with the neighbors.
- If a family f contains 3 persons with weights $w_1^f \leqslant w_2^f \leqslant w_3^f$, we set $f_{\text{tot}} = w_1^f + w_2^f + w_3^f$, and $f_{\min} = w_1^f + w_2^f$, since f_1 and f_2 will crash twice.
- If a family f has size 2, then its two members, with weights $w_1^f \leqslant w_2^f$, will play both the role of f_1 and f_2, and the one of f_3 and f_4, respectively, because they will both crash twice. That is, we must set $f_{\text{tot}} = 2(w_1^f + w_2^f)$.
- Finally, if a family f has size 1, its unique member f will necessarily be an antenna–person, and hence its contribution to the total cost might be high if the weight w^f of such person is high; in order to count this we must set $f_{\text{tot}} = 2w^f$ and $f_{\min} = w^f$.

The algorithm above leads to an optimal solution of cost $\bar{\mu}$ because, no matter of how we place the rectangles resulting from *Step* 1, the cost μ_f of each family is minimized, i.e., we have the following:

$$\min \mu_f = \begin{cases} f_{\text{tot}} & \text{if } f \in \mathbf{F}_{\text{even}}, \\ f_{\text{tot}} + f_{\min} & \text{if } f \in \mathbf{F}_{\text{odd}} \end{cases}$$

and the total cost of crashes of an optimal room allocation ρ will be

$$\bar{\mu} = \min \mu(\rho) = \sum_{f \in \mathbf{F}} f_{\text{tot}} + \sum_{f \in \mathbf{F}_{\text{odd}}} f_{\min}. \tag{1}$$

In fact, each one of the four minimal-weighted members of each family of size at least four will have a crash, and it will contribute with its weight to the total cost. Moreover, for odd size families, the least important member will add once more its weight to the cost as it will crash twice with neighbors. The definition of f_{tot} and $f_{\min}$, extended to families with less than four members as shown above, guarantees the correctness of the approach.

The procedure introduced in this section gives the intuition of our strategy, which is based on the optimal substructure of the problem, and that results in a linear time algorithm for an instance of the QAP (the complexity analysis will follow). In the next section we further explore the room allocation problem relaxing the unrealistic condition of the circular shape of the corridor, and we show that a linear time algorithm still exists.

3.2. A rectangular corridor

Let us now consider the corridor of Fig. 1. The only difference with the circular corridor lies in the existence of the two side borders, which allow to save some crashes for the families placed there. This possibility requires a preprocessing phase for determining the *border families* (that is, families to be placed at the borders), and attention to special cases that might rise as a consequence. After this preprocessing phase, consisting of two steps, we basically apply the strategy described in the previous section.

We first consider the case of families with at least four members.

The first step consists in detecting the most suitable families for the two borders of the corridor. In order to minimize the crash costs, we select for the borders the two families (one per border) with the two highest values $f_{\rm b}=w_3^f+w_4^f$. This is indeed the cost that is saved w.r.t. the case of a circular corridor.

Now, depending on the size of the families placed at the borders, we can have four different cases

(1) If the two families placed at the borders have even size, the remaining families can be (paired and) placed as in the case of the circular corridor leading definitely to an optimal solution.
(2) If only one of the border families has odd size, we can pair it with another odd family (which certainly exists, since, by hypothesis, $n=2m$ and we have an even number of odd size families), and we proceed as in case (1).
(3) If both border families have odd size, and we have at least four odd size families, we pair each border family with another odd size family, and proceed as in case (1).
(4) If there are only two odd size families, and they are both selected for the borders, then we are left with a non-rectangular corridor, and we cannot proceed as in case (1) because only even size families (hence rectangles) are left.

Hence, in the first three cases, the strategy explained in the previous section guarantees an optimal solution, while case (4) has to be dealt differently.

The second step of the preprocessing is in fact meant to deal with case (4) of the first step.

There are basically two alternative strategies. The first is to give up about keeping a rectangular shape for even size families, and the second is to split one even family into two-odd size subfamilies and proceed as in case (3) of the first step. In the first strategy, in fact, there is no way to place an even size family against the odd size family at the border without resulting in a shape (for the remaining corridor) which has antennae on both sides. By iterating this up to the other border, the result is that each even size family shape has one antenna per side.[2]

In the second strategy, a suitable choice of the family to be split has to be made.

Hence, we now concentrate on the costs of the two resulting allocations, in order to determine which strategy leads to an optimal solution. In the first strategy, each even size family f brings an extra cost due to the fact that it has two antenna elements so that the weights w_1^f and w_2^f will actually be counted twice. Hence, the cost of the resulting allocation would be $c_1=\bar{\mu}+\sum_{\text{even } f\in\mathbf{F}}(w_1^f+w_2^f)-f_{\rm b}^{\rm o1}-f_{\rm b}^{\rm o2}$, where $f^{\rm o1}$ and $f^{\rm o2}$ are the two odd size families and $\bar{\mu}$ is defined as in Eq. (1). Concerning the second strategy, we first observe that it is *always* convenient to split a family f in two parts where one part contains a single element f_1 crashing three times, rather than leaving at least three elements at each side. In fact, the latter case would introduce new crashes (definitely heavier than w_1^f) while saving only one (out of three) crash to f_1. Hence, any split (possibly also involving one of the two border odd size families) should be done isolating only one element. Summing up, splitting one family f allows to place all unsplit even size families in rectangular shape and thus to save the costs brought by two elements being antennae for all of them. On the other hand, the split of f causes $\mu_f(\rho)>f_{\rm tot}$. Naming $c_f=\mu_f(\rho)-f_{\rm tot}$ such extra cost, it can be verified that

$$c_f=\begin{cases}2w_1^f+w_2^f+w_5^f & \text{if } f\in\mathbf{F}_{\rm even} \text{ and has at least 6 elements},\\ 2w_1^f+w_2^f+w_3^f & \text{if } f\in\mathbf{F}_{\rm even} \text{ and has 4 elements},\\ w_1^f+w_3^f & \text{if } f\in\mathbf{F}_{\rm odd}.\end{cases}$$

Formally, the overall cost (which depends on the choice of f) is $c_2(f)=c_1-\sum_{\text{even } g\in\mathbf{F}}(w_1^g+w_2^g)+c_f$. As a consequence, the choice between the first and the second strategy can be made by just comparing c_1 and c_2, where $c_2=\min_{f\in\mathbf{F}}c_2(f)$. Notice that it is never convenient to give up about placing the two odd size families at the borders. In fact, the arguments about the possibly split family also hold for any of the two border families, and it can be checked that it is always convenient to perform such split rather than pairing the two odd size families into a rectangle.

We finally consider the general case with families of any size. First of all we define $f_{\rm b}$ for families with at most 3 members: if family f contains 3 persons, then $f_{\rm b}=w_2^f+w_3^f$; if f has size 2, then $f_{\rm b}=w_1^f+w_2^f$; and if f is of size 1 we have $f_{\rm b}=w^f$. For families of size 2 and 3, the assignments to the parameters given above are enough to let them be optimally placed using the algorithm as for the case of families of size 4 or more. On the other hand, families of size 1 require particular care concerning their placement at the borders of the corridor, as this implies a second family being placed at the same border of the corridor. This issue is addressed in the following proposition, where we assume that there is only one family u of size 1 to be placed at one border, and we characterize the other families to be placed at the borders. Of course, for family u, $u_{\min}=u_{\rm b}=w^u$. We also assume that we are not in the case where there is only one odd

[2] It is easy to verify that any other shape results in more crashes.

size family other than u, and that they are both to be placed at the borders (this case has been already addressed earlier in this section).

Proposition 2. *Let u be a family of size 1, such that u_b is maximum (with respect to the value of t_b of each family t), let $f \in \mathbf{F}_{odd}$ with f_4 maximum, and let g be the family with the third maximum value of g_b. Finally, let h be the one with the second maximum value of h_b.*

We have that in the optimal solution, one border will be occupied by the family h, while for the other border we can have two possibilities. If

$$u_{min} + f_{min} \geqslant g_b,$$

then u and f are placed at the other border in the following way:

$$\left| [u][f][f] \right.$$

$$\left| [f][f][f] \right.$$

Else, g will be the second border family.

Proof. Let us first consider the case, where f and g are the same family. There are two possible allocations:

(1) The first allocation (with f_4 in the antenna room) is

$$\left| \begin{array}{ll} [u][f_3]\cdots & [f_2] \\ [f_4]\cdots\cdots & [f_1] \end{array} \right.$$

and its cost, i.e., its contribution to $\mu(\rho)$, is given by $f_{tot} + 2u_{min}$.

(2) The second possible allocation (with f_1 in the antenna room) is

$$\left| [f_4]\cdots \quad \cdots[f_1]\cdots \quad \cdots \right.$$

$$\left| [f_3]\cdots[f_2]\cdots \quad \cdots[u] \right.$$

and its cost is given by $2w_1^f + w_2^f + 3u_{min}$.

Thus the thesis directly follows by comparing the costs of the two possible allocations.

Now suppose that f and g are two different families, with g an even size family (the odd size case is analogous). We have to compare the cost of the two possible allocations.

(1) The first allocation is

$$\left| \begin{array}{llll} [u][f_3]\cdots & [f_2]\cdots \quad \cdots & [g_1]\cdots & [g_2]\cdots \\ [f_4]\cdots \quad \cdots & [f_1]\cdots \quad \cdots & [g_3]\cdots & [g_4]\cdots \end{array} \right.$$

and its cost is given by $f_{tot} + 2u_{min} + g_{tot}$.

(2) The second possible allocation is

$$\left| [g_4]\cdots[g_1]\cdots \quad \cdots \quad \cdots[f_2]\cdots[f_3]\cdots \quad \cdots[u]\cdots \right.$$

$$\left| [g_3]\cdots[g_2]\cdots \quad \cdots[f_1]\cdots \quad \cdots[f_4]\cdots \quad \cdots \quad \cdots \quad \cdots \right.$$

and its cost is $f_{min} + f_{tot} + 3u_{min} + w_1^g + w_2^g$.

Thus the thesis directly follows (note that $g_{tot} - (w_1^g + w_2^g) = g_b$). $\square$

This result can be extended to the cases of two, three, or four size 1 families that maximize f_b. The number of possibilities to be evaluated increases, but it remains a constant and, thus, it does not change the overall complexity of the algorithm.

3.3. Complexity analysis

We now show that the algorithm we suggested takes a time that is linear in the number n of people. We have, say, p families each one having size n_i with $i = 1, \ldots, p$ and $n_1 + \cdots + n_p = n$. Searching within the ith family the 4 members with lower weight can clearly be done in $O(n_i)$ time. Assigning values to the parameters f_b, f_{tot}, f_{min}, and possibly c_f, can be done at the same time on the fly. This gives an overall $O(n)$ time for computing them all. In order to generate an optimal solution, one has just to assign families to the borders (which takes linear time), and then make the random choice of how to place the rest, which can be done in $O(n)$ time. Notice that the preprocessing introduced in Section 3.2 does not affect the linearity of the algorithm. In summary, finding a room allocation ρ which minimizes $\mu(\rho)$ takes linear time. It is straightforward to see that also the space complexity is linear.

4. A corridor with an extra room

When n is an odd number (that is $n = 2m - 1$), then there is an empty room left, which can be used to avoid three conflicts. Indeed, we should use the empty room to transform an odd size family into an even size one. Of course, we could also transform an even family into an odd one, but it is easy to observe that this is not convenient. Thus, let g be an odd size family, and f be any family. We can allocate the empty room as follows:

$$
\begin{array}{l}
[g_2]\cdots[g_4][\quad][f_4]\cdots \\[1em]
[g_3]\cdots\ \cdots[g_1][f_3]\cdots
\end{array}
$$

and the total gain is $w_4^g + w_1^g + w_4^f$. Therefore we have to choose the odd size family g that maximizes $w_4^g + w_1^g$ and, among the families left, the one with w_4^f maximum. Note that we can also have the following allocation

$$
\begin{array}{l}
\cdots[g_4][u]\ [f_4]\cdots \\
\cdots[g_3][u]\ [f_3]\cdots
\end{array}
$$

where u is the unique member of a family of size one. In this case, the total gain is $w_4^g + w_u + w_4^f$, which should be compared with the previous value in order to choose the most convenient solution.

The problem of room allocation in a corridor with an extra room can be formulated as a QAP as well. The QAP for this case can be obtained from the one described in Section 2 modifying the first constraint as $\sum_{i=1}^{n} X_{ij} = y_j$ with $y_j \in \{0, 1\}$, and adding the new constraint $\sum_{j=1}^{2m} y_j = n - 1$.

References

[1] D. Adolphson, T.C. Hu, Optimal linear ordering, SIAM J. Appl. Math. 25 (3) (1973) 403–423.

[2] N. Christofides, M. Gerrard, Special cases of the quadratic assignment problem, Management Science Research Report 391, Carnegie Mellon University, 1976.

[3] V. Ciriani, N. Pisanti, A. Bernasconi, Efficient optimal greedy algorithms for room allocation, Proceedings of the Second International Conference Fun with Algorithms, 2001, pp. 43–59.

[4] A. Frangioni, Solving semidefinite quadratic problems within nonsmooth optimization algorithms, Comput. Oper. Res. 23 (11) (1996) 1099–1118.

[5] G. Gallo, B. Simeone, Optimal grouping of researchers into departments, Ricerca Operativa 57 (1991) 45–69.

[6] F. Maluccelli, A polynomially solvable class of quadratic semi-assignment problems, Eur. J. Oper. Res. 91 (1996) 619–622.

[7] P.N. Pardalos, F. Rendl, H. Wolkowicz, The quadratic assignment problem: a survey and recent developments, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 16 (1994) 1–41.

[8] F. Rendl, Quadratic assignment problems on series-parallel digraphs, Z. Oper. Res. 30 (1986) 161–173.